**JENKINS:**

1. creating & Managing jobs
   1. Go to Jenkins Dashboard->New item->select the job type(freestyle, pipeline,Maven , Multibranch pipeline etc)
   2. Then enter the pipeline script r clone the git repo from SCM. Configure the job
   3. Once the pipeline created, save the job configuration by clicking on **“Save”** at bottom of the page.
   4. Then build the job by running the job on selecting “**build now”** option.
   5. We can be able to view the build result from the console output page.
2. types of Jenkins jobs and its differences
3. Freestyle job – Allows to create custom build and deployment workflows.
4. Pipeline job used to create pipelines mainly we used for the deployment of an application and automation .
5. Multibranch pipeline –. Check all the branches in the git repo and run each build for each Jenkins file present in different branches
6. Maven project – Used for building s/w projects using Apache Maven.
7. It automates the tasks like compiling source code, running tests cases and generating reports based on maven config file (pom.xml)
8. o/p – under “target” folder
9. creating & adding credential in Jenkins and its types

**info**

1. **Tools (git , ansible tower , nexus , jenkins , ), APIs , AWS services (ex- ECR , Code artifact) & servers ,**
2. **Individual(own – employee id , password [windows login]) – dynamic(expired in 3 months) and service account (ex- HK-SVC-HAP . password – 27hdyr93) 🡪 static**

**Used to authenticate & authorize users to**

Jenkins Dashboard->Manage Jenkins-> Credentials (mostly manage jenkins will not be seen in your Jenkins at orgz level)

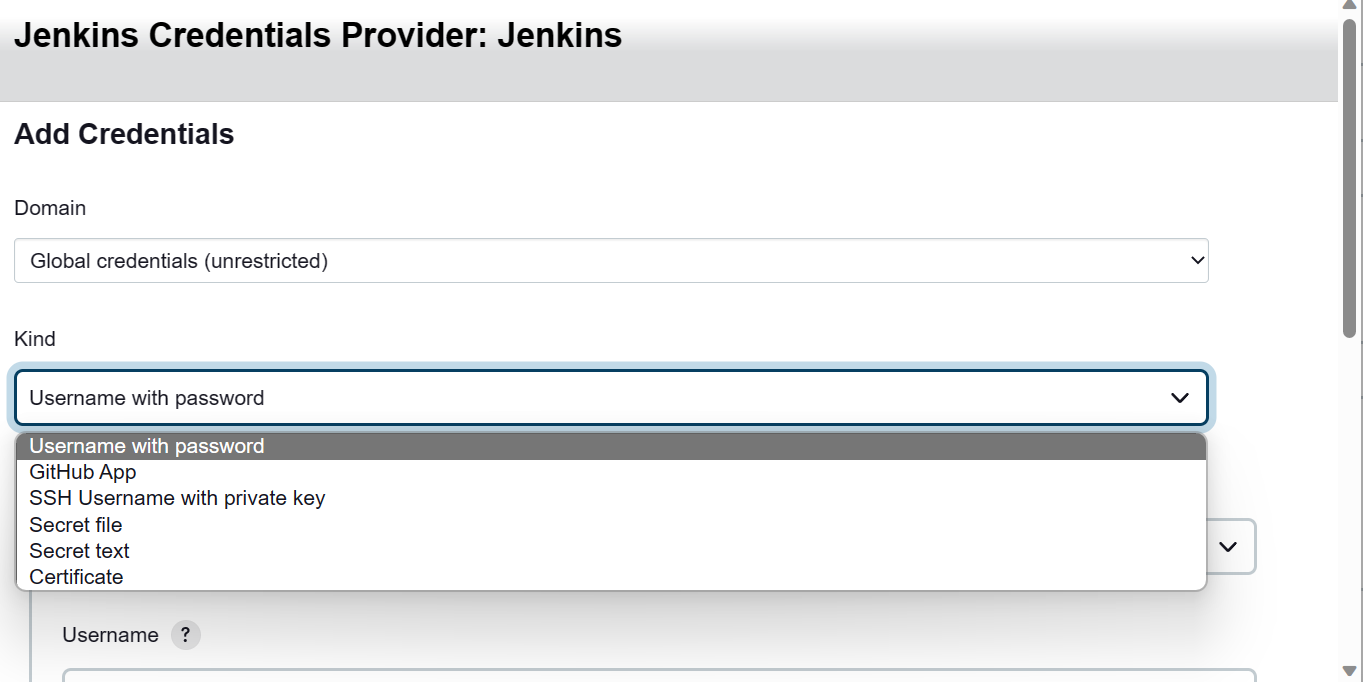
* + - 1. Under system->Global credentials link will be there to access default domain system.
      2. Click **“Add Credentials”** and provide details like **type (usrname with pwd, SSH username & private key, Secret file, secret text, etc), Scope (Global r system).**

**Types**

1. Username & pwd – Specifies the username & pwd. (git,Jenkins,tower,nexus etc)
2. Secret text – such as API token. (API)
3. Secret file – secret content in a file.
4. SSH username with private key – Enter SSh username and private key for authentication.

# What is a login credential?

* Login credentials are managed usernames and passwords with access to various applications.
* Various Types of Jenkins Credentials?
* Username with Password
* GitHub App
* SSH username with private key
* Secret file
* Secret Text
* Certificate



While Creating a job we can assign the credential with respective usage of job.

1. what is parameters & its types
2. Used to pass values to Jenkins pipeline code during runtime

**Types**

* + 1. **Boolean –** have a value of either true / false. Used to enable / disable features in Jenkins
    2. **Choice –** Allows user to select value from a predefined list of options. – ( ex- servers list – adding all the deployment servers)
    3. **String –** Allows user to enter a text value during runtime (ex- giturl , branch , etc )
    4. **Multi-line string –** Allows use to enter multi-line parameter (ex- multi line texts)
    5. **File –** Allows user to upload a file during runtime

1. Build in environmental variables
2. These are the default variables in jenkins which values can’t be changed

For Eg:

Env.BUILD\_ID 🡪 Unique identifier for current build

Echo “${Env.BUILD\_ID }” 🡪 it will print the current build id

Env.BUILD\_NUMBER 🡪

Echo “${Env.BUILD\_NUMBER }” 🡪 it will print the current build number

Env. JOB\_NAME -> 🡪 it will print the job name

Env.Build\_User 🡪 print the person name who runs the pipeline

BUILD\_TAG 🡪 tag name for current build

JENKIN\_HOME 🡪 path to jenkin home directory

WORKSPACE 🡪 path to workspace directory for current build

**Example script:**

**pipeline {**

**agent any**

**environment {**

**MY\_VARIABLE = "Hello, world!"**

**}**

**stages {**

**stage('Example Stage') {**

**steps {**

**echo "My variable value is: ${env.MY\_VARIABLE}"**

**}**

**}**

**}**

**}**

1. scm checkout

Cloning the git repo to Jenkins workspace

1. In jenkin job page->Under pipeline section select **“Pipeline script from SCM”**
2. Select Git under SCM field.

Give Git URL and credentials, branch, script path

1. build triggers & types

**Webhook method** - If there any changes in git repo, automatically alerts the jenkins to run build

**Build periodically** - Provides **cron** like (adding time) feature to periodically execute the project.

**Poll SCM** – Provides **cron** like (adding time) feature to periodically execute the project but it will only if there is new commit in the git repo

* + - 1. It uses **cron** syntax (5 astrixs)
    1. Minutes, hours, day of the month, month, day of the week

**Trigger Remotely** – A token will be generated , by using this anyone who doesn’t have access to the Jenkins also can trigger the job

1. email notification

Step 1:

* Manage Jenkins -> Configure System-> E-mail Notification
* In E-mail Notification Select SMTP server
* In SMPT server provide your Mail SMPT Server (ie, if it’s a Gmail default SMPT server is “smpt.gmail.com”)
* Then, go to Advance section-> Select SMTP authentication

Provide Username as your mail id

And for Password

: Goto your Gmail🡪 Select security🡪 select verify with 2step authentication🡪Complete the 2 step authentication with your Mobile OTP

: Then After Select APP password 🡪 Generate a new App Password in the name of Jenkins

: Copy the Password

* In SMTP authentication 🡪 give username as your mail id and password is your newly generated app password
* Then check the box SSL
* Provide the SMTP Port no (default for your Gmail smtp port number is 465)
* And reply to the address provide your mail id
* Other section leave as it is.

Step 2:

* Create a freestyle Job
* Give the sh script
* Below the script line select postbuild option
* In post build option 🡪 select E-mail notification 🡪 Provide the details
* You can able to get the e-mail notification for a particular job.

# Extended E-mail Notification

* Install the plugin in the name of Extended e-mail notification
* Once install 🡪 manage Jenkins 🡪 configure system🡪Select Extended E-mail notification
* Same Configuration has to be done as per the below configuration as E-mail notification
* Then go to job🡪 in that select post build option 🡪 Select Extended E-mail notification
* In this Extended E-mail Notification, we can edit your e-mail content.
* Difference between E-mail notification and Extended E-mail notification is in Extended E-mail notification, We can custom your E-mail as much you want

1. how to run shell script file using execute shell option in Jenkins (to run a shell script file - “./filename “)
2. Create a new job freestyle Jobs
3. Select Build Step as Execute shell

A screenshot of a computer program

Description automatically generated with medium confidence

1. Write a Shell commands

A screenshot of a computer

Description automatically generated with low confidence

1. Save the Project and Build the job
2. The console output will be

A computer code on a white background

Description automatically generated with low confidence

1. how to run shell script file using pipeline script and also linux commands

**using pipeline script:** we try hello world script

Pipeline {

Agent any

Stages {

Stage(‘Hello’) {

Steps {

Sh ‘’’path/to/your/script.sh’’’

}

}

}

}

1. difference between declarative and scripted pipeline and syntax

* What is Scripted pipeline in Jenkins?
* based on Groovy scripting language.
* Tradition and complex

# What is declarative pipeline in Jenkins?

* based on the Groovy programming language
* Newly added and simpler syntax
* it uses a YAML-based syntax for defining the pipeline

Example of Declarative Pipeline:

pipeline {

agent any // or agent(‘machine1’) //any – runs on any available machine // specifying name(machine1) – job runs on particular machine

stages {

stage('Build') {

steps {

//

}

}

stage('Test') {

steps {

//

}

}

stage('Deploy') {

steps {

//

}

}

}

}

Example of Scripted Pipeline:

node {

stage('Build') {

//

}

stage('Test') {

//

}

stage('Deploy') {

//

}

}

1. creating groups and adding user to groups - access

Mostly below steps are done by Jenkins admin team

Creating groups and adding user to groups - access

Go to Manage Jenkins

Go to Configure Global Security.

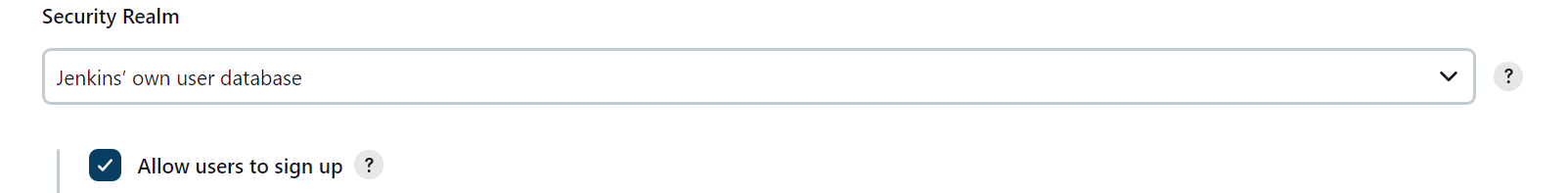
A screenshot of a computer

Description automatically generated with low confidence

Select Jenkins own User Database.



Tick the checkbox “Allow Users to Sign Up”



Select Matrix-based security in Authorization section

A picture containing text, screenshot, line, font

Description automatically generated

Select the button “Add user or group

A screenshot of a computer

Description automatically generated with low confidence

Save and Apply.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

In real -time

Roles – add roles – name and (new – it has build access)

provide access (build , build & configure , admin)

Groups – add groups – name , adding the role (hpp – adding the ‘new’ role)

Add Users – under group and add the created role (users adding under ‘hpp’ group)

1. pipeline syntax - snippet generator

Instead of writing the code Jenkins provide us an option which will help to generate the code as per our need

For ex- for scm checkout(git cloning)we can easily generate the code using this option

* In the job build page 🡪 select pipeline syntax at the bottom of pipeline section.
* Select snippet generator 🡪 select GIT in sample step field 🡪 Provide repo URL, branch, cred and generate.
* O/P – pipeline code will be generated
* Then, copy & paste the code to pipeline script section.

1. shared library
2. master slave node connectivity

**Summarized steps**

1. **Key transfer**
   1. Copy the generated key from master and put it in slave (.ssh/authorized\_files)
2. **Jenkins** –
   * 1. Configuring nodes – name , executors , slave ip , Launch agent via SSH , Adding credential and host verification strategy

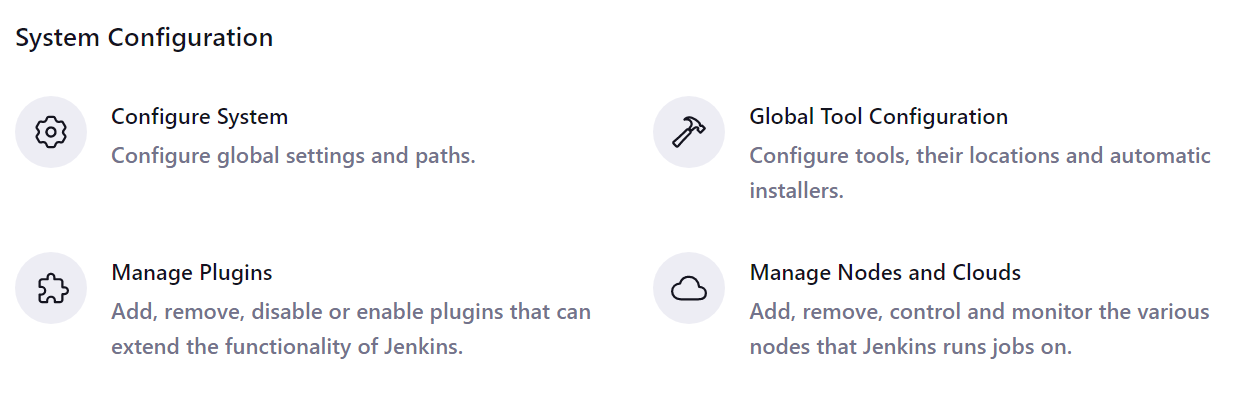
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

# Master slave configuration,Node Level

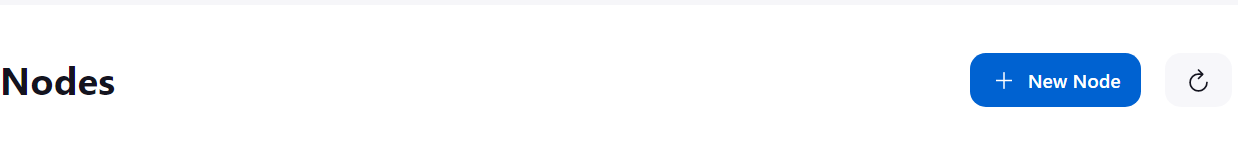
Step’s

**:** Logon to Jenkins

**:** Manage Jenkins 🡪 Manage Nodes and Clouds

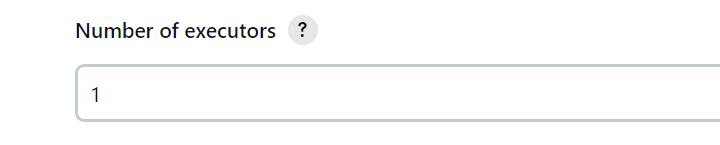


: Create New node

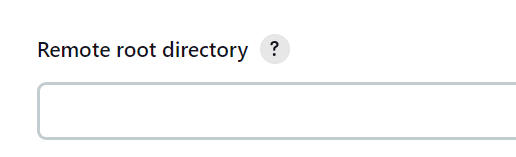


: Give the node name and selected as a Permanent Agent

: Select number of Executors you want



: Provide the IP address of the slave machine in Remote root directory



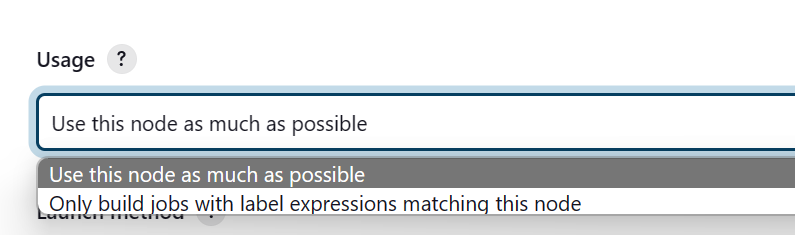
: Give the lable name to identify the node’s

A picture containing text, screenshot, font, line

Description automatically generated

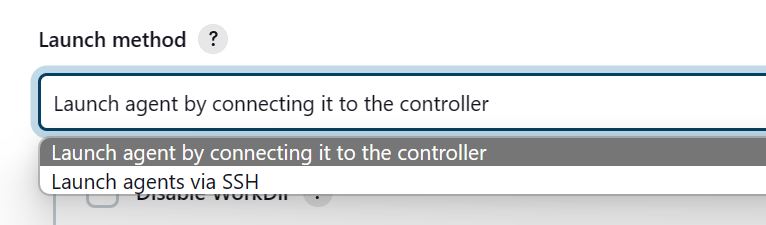
: Usage of node

* Select Use this node as much as possiable

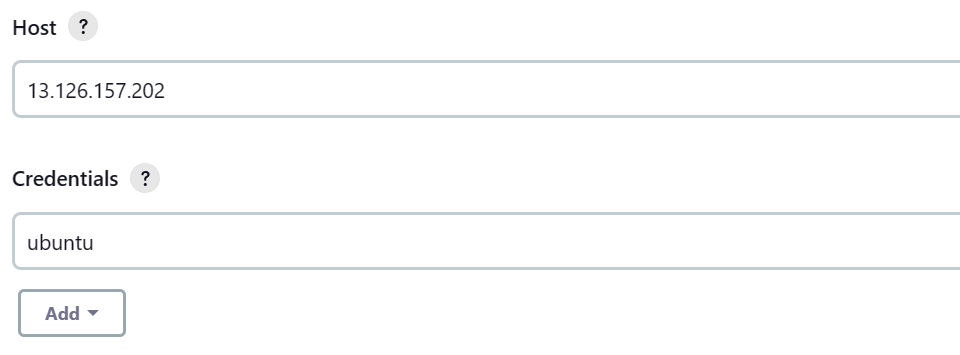


: Launch Method

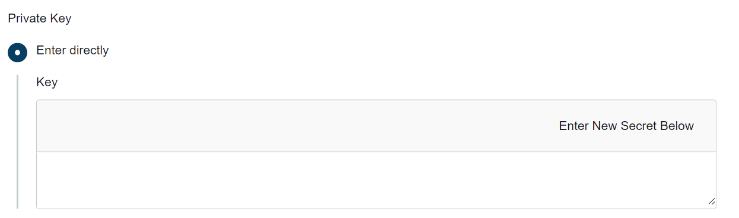
* Select Launch agents via SSH



:

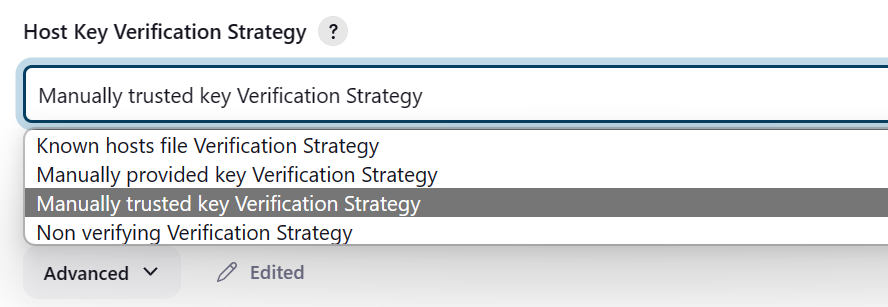
* Provid the Ip address of your slave machine
* 

: Create a Credentials with Jenkins

* Select Domain as “ Global Credentials(unrestricted)”
* Select Kind as “ SSH username with Private key
* Scope as it is
* Provide ID
* Give Username of your slave machine
* IN private key
  + - 1. Select ADD to add the private key
* Private key is your Slave machine .pem file
  + - * Open pem file in Notepad copy the content .
      * Add it to here
      * 

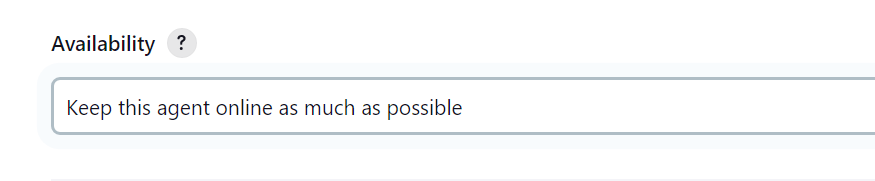
: Host Key Verification Strategy

* Select Manually Trusted Verification Strategy



: Availability

* Select Keep this agent online as much as possible.

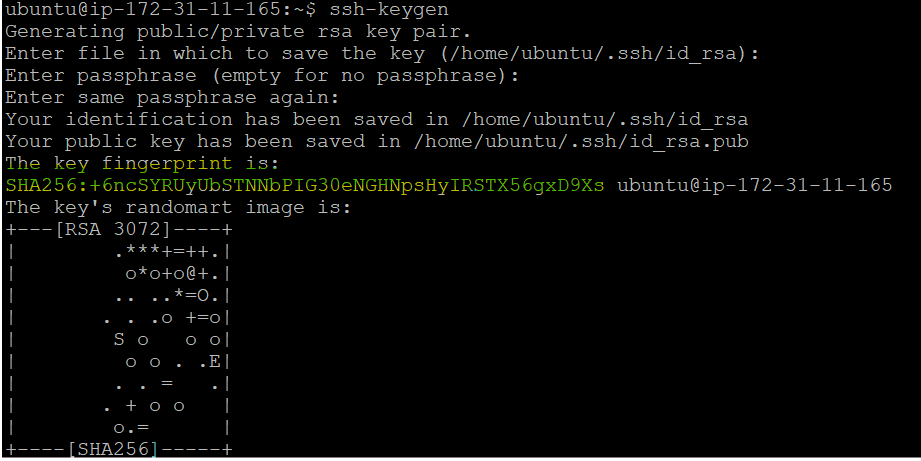


: Save the Node

## In your master machine

: Generate the ssh key

By using the command ssh-keygen



: Copy the key and paste it in your slave machine Authorized key

* /home/ubuntu/.ssh
* Open Authorized key and past the key in it.

: Now Create a Job in Jenkins and run that job in your salve machine

1. various Jenkins job features